Elektrotehnički fakultet

Univerzitet u Istočnom Sarajevu



Predmet: Paralelni računarski sistemi

Tema: Paralelizam u algoritmima sortiranja – Quick sort i Shell sort

Autori: Milan Vlaški i Fejzullah Ždralović

Sadržaj

1. Uvod 3

1.1 Svrha 3

1.2 Definicije, akronimi i skraćenice 3

1.3 Reference 3

1.4 Pregled 3

2. Paralelizam 3

2.1 Razlika između paralelizma i konkurentnosti 3

2.2 O paralelizmu 4

2.3 Prednosti paralelizma 4

2.4 Nedostaci paralelizma 5

3. Java 5

3.1 O tehnologiji 5

3.2 Verzije Jave 6

3.3 Instalacija Jave 6

3.4 Biblioteke 7

4. Sortiranje 7

4.1 Algoritmi sortiranja 7

4.2 Quick sort 7

4.3 Shell sort 7

5. Paralelizam u algoritmima sortiranja 7

6. Paralelizam u Javi 7

# Uvod

[O dokumentu.]

## Svrha

[Šta želimo postići ovim radom, a i ovim dokumentom**.**]

## Definicije, akronimi i skraćenice

[Sve riječi, iz domena problema, koje je potrebno detaljnije objasniti, potrebno je navesti ovjde.]

* JVM – Java Virtuelna Mašina
* IDE - Integrated Development Environment (Integrisano razvojno okruženje)

## Reference

[Izvori koje smo koristili prilikom pisanja rada.]

* <https://wiki.haskell.org/Parallelism_vs._Concurrency>
* "Encyclopedia of Parallel Computing" by David Padua
* "Concurrency: State Models & Java Programs" by Jeff Magee and Jeff Kramer
* <https://www.geeksforgeeks.org/introduction-to-parallel-computing>
* https://www.javatpoint.com/java-versions

## Pregled

[Pregled čitavog dokumenta. Šta sve on sadrži.]

# Paralelizam

## Razlika između paralelizma i konkurentnosti

Da ne bi došlo do zabune, na samom početku, napravit ćemo razliku između dva pojma, a to su paralelizam i konkurentnost. Iako su paralelizam i konkurentnost povezani, to su ipak različiti koncepti u računarstvu.

Paralelizam se odnosi na upotrebu više procesora ili procesorskih jezgara za obavljanje zadataka istovremeno kako bi se povećala računska efikasnost i smanjilo vrijeme obrade. Paralelizam uključuje podjelu većeg zadatka na manje podzadatke koje mogu paralelno obraditi različiti procesori ili jezgra za obradu. Glavni cilj paralelizma je postizanje brže i efikasnije obrade zadatka.

Konkurencija se odnosi na sposobnost različitih dijelova sistema da se izvršavaju nezavisno i istovremeno. Konkurencija je sposobnost programa da istovremeno obrađuje više zadataka ili zahtjeva, bez čekanja da se jedan zadatak završi prije nego što započne drugi. Glavni cilj konkurentnosti je poboljšanje ukupne odzivnosti i efikasnosti sistema. Na primjer, to može biti izvršavanje više zadataka na jednoprocesorskoj mašini.

Na slikama ispod ćemo predstaviti paralelizam i konkurentnost, kako bismo ih bolje shvatili i zapamtili.



Slika 1



Slika 2

## O paralelizmu

[Pišemo o tome šta je paralelizam. Kako je došlo do njega. Koje su njegove prednosti, a koje mane.]

Termin paralelizam odnosi se na tehnike za ubrzavanje programa izvođenjem nekoliko računanja u isto vrijeme. To je zapravo upotreba više procesora ili procesorskih jezgara za istovremeno obavljanje zadataka kako bi se povećala računska efikasnost i smanjilo vrijeme obrade. Zbog toga je potreban hardver sa više procesorskih jedinica. U mnogim slučajevima, potproračuni su iste strukture. Paralelizam se postiže paralelnom obradom, što uključuje podjelu većeg zadatka na manje podzadatke koje mogu paralelno obraditi različiti procesori ili jezgra za obradu. Na primjer, Grafička izračunavanja na GPU-u (Graphics processing unit) su paralelizovana. Paralelizam se može koristiti u različitim računarskim aplikacijama, uključujući naučne simulacije, analizu podataka, obradu slika i videa i umjetnu inteligenciju.

## Prednosti paralelizma

Prednosti paralelizma uključuje povećanu brzinu i efikasnost, poboljšane performanse i skalabilnost računarskih sistema, uštedu troškova, veću fleksibilnost, poboljšanu toleranciju na greške, bolje korištenje resursa, poboljšana produktivnost, brža inovacija . U nastavku ćemo nešto reći o svakoj od ovih prednosti.

*Povećana brzina i efikasnost*: paralelizam omogućava da više procesora ili procesorskih jezgara rade zajedno, u isto vrijeme, što može značajno povećati brzinu i efikasnost obrade zadataka.

*Poboljšane performanse i skalabilnost*: Paralelizam može poboljšati performanse i skalabilnost računarskih sistema, čineći mogućim rukovanje većim i složenijim zadacima i skupovima podataka.

*Uštede troškova:* Korištenjem paralelne obrade za poboljšanje efikasnosti obrade, organizacije mogu potencijalno uštedjeti na troškovima hardvera, energije i ukupnih troškova računarske infrastrukture.

*Veća fleksibilnost*: Paralelizam može pružiti veću fleksibilnost u smislu načina na koji se računski resursi dodjeljuju i koriste, omogućavajući organizacijama da prilagode resurse obrade prema potrebi kako bi zadovoljili promjenjive zahtjeve.

*Poboljšana tolerancija grešaka*: Korištenjem redundantnosti i mehanizama za pronalazak greške, paralelizam može poboljšati toleranciju grešaka i pouzdanost računarskih sistema, smanjujući rizik od kvarova sistema i zastoja.

*Bolje korištenje resursa*: Paralelizam omogućava efikasno korištenje dostupnih računarskih resursa omogućavajući istovremeno pokretanje više procesa, smanjujući vreme mirovanja i poboljšavajući ukupnu iskorišćenost sistema.

*Poboljšana produktivnost*: Smanjenjem vremena obrade i povećanjem efikasnosti računarskih sistema, paralelizam može omogućiti organizacijama da obavljaju zadatke brže i efikasnije, povećavajući produktivnost i propusnost.

*Brža inovacija*: Paralelizam može omogućiti bržu inovaciju omogućavajući istraživačima i programerima da obrađuju velike skupove podataka i izvode složene simulacije brže i efikasnije, što dovodi do bržih uvida i otkrića.

## Nedostaci paralelizma

Jedan od ključnih problema sa paralelizmom je pitanje upravljanja i koordinacije različitih paralelnih procesa kako bi se osiguralo da oni rade zajedno efikasno i bez sukoba. Ovo može biti posebno izazovno u distribuiranim računarskim sistemima, gde se procesi mogu izvoditi na različitim mašinama povezanim mrežom.

Drugi ključni problem je pitanje balansiranja opterećenja, koje uključuje ravnomjernu raspodjelu radnog opterećenja na različite procesorske jedinice kako bi se osiguralo da nijedna jedinica ne miruje dok su druge preopterećene. Balansiranje opterećenja može biti posebno izazovno kada je radno opterećenje dinamično i nepredvidivo, jer može zahtijevati stalno prilagođavanje distribucije zadataka na osnovu promjenjivog radnog opterećenja.

Ostala pitanja vezana uz paralelizam uključuju potrebu za efikasnom komunikacijom i sinhronizacijom između paralelnih procesa, osiguravajući da se podaci pravilno dijele i ažuriraju između procesa i rješavanje potencijalnih uvjeta utrke, zastoja i drugih problema povezanih s konkurentnošću.

Ključni problem paralelizma je smanjenje ovisnosti podataka kako bi se mogli izvoditi proračuni na nezavisnim računskim jedinicama uz minimalnu komunikaciju između njih. U tu svrhu, čak može biti prednost da se isto izračuna dva puta na različitim jedinicama

# Java

[Objašnjavamo tehnologiju u kojoj ćemo raditi projekat. Govorimo o verziji koju upotrebljavamo, o tome odakle smo preuzeli instalaciju, o bibliotekama koje smo koristili]

Java je objektno orijentisani programski jezik visokog nivoa, koji je prvi put objavio Sun Microsystems 1995. Danas je u vlasništvu Oracle kompanije. Dizajniran je tako da bude nezavisan od platforme, što znači da se Java kod može kompajlirati i pokrenuti na bilo kojoj platformi koja ima Java virtuelnu mašinu (JVM), što ga čini veoma prenosivim.

## O tehnologiji

Spomenut ćemo neke od ključnih karakteristika Java programskog jezika.

Java je potpuno objektno orijentisani programski jezik, što znači da se svo programiranje vrši korištenjem objekata i klasa. Java posjeduje ugrađen Garbage Collector, koji koristi automatsko upravljanje memorijom, što olakšava upravljanje memorijom i smanjuje rizik od curenja memorije. Već smo spomenuli da je Java nezavisna od platforme, pa se Java kod kompajlira u bajt kod, koji se može pokrenuti na bilo kojoj platformi koja ima instaliranu Java virtuelnu mašinu (JVM), bez obzira na osnovni hardver ili operativni sistem. Java ima ugrađenu podršku za višenitnost, koja omogućava da se više niti izvršava istovremeno i asinhrono. Java ima snažan sigurnosni model koji uključuje funkcije kao što su provjera bajt koda i sandboxing za zaštitu od zlonamjernog koda. Java programski jezik ima veoma dobre mogućnosti rukovanja izuzecima, koje olakšavaju pisanje koda koji može rukovati neočekivanim greškama ili izuzecima na zadovoljavajući način. Java ima mnogo popularnih IDE-a (Integrisano razvojno okruženje) kao što su Eclipse, IntelliJ IDEA i NetBeans koji pružaju funkcije kao što su dovršavanje koda, otklanjanje grešaka i refaktorisanje, što olakšava razvoj Java aplikacija.

Java se široko koristi u raznim aplikacijama, uključujući web razvoj, razvoj mobilnih aplikacija, razvoj poslovnog softvera i naučno računarstvo. Ima veliku i aktivnu zajednicu programera i korisnika, što je doprinijelo razvoju bogatog sistema biblioteka, frajmvorka i alata.

## Verzije Jave

[bla, bla, bla]

Java je imala zavidan broj verzija i izdanja od svog prvog izlaska 1995. godine. U nastavku smo nabrojali neka od glavnih Javainih izdanja:

Java 1.0: Prvo stabilno izdanje Jave, predstavljeno 1996. godine.

Java 1.1: Uvedeno 1997. godine, ovo izdanje je dodalo podršku za unutrašnje klase i JavaBeans.

Java 1.2 (takođe poznat kao Java 2): Uvedeno 1998. godine, ovo izdanje je dodalo podršku za Swing GUI alat, Java imena i interfejs imenika (JNDI) i Java Foundation Classes (JFC).

Java 1.3: Objavljena 2000. godine, ova verzija je uvela Java Sound API i podršku za HotSpot JVM.

Java 1.4: Objavljena 2002. godine, ova verzija je uvela Java Native Interface (JNI), XML obradu sa JAXP-om i regularne izraze sa Java.util.regex.

Java 5 (takođe poznat kao Java 1.5): Objavljeno 2004. godine, ovo izdanje je dodalo podršku za generičke karakteristike, napomene i autoboxing.

Java 6 (takođe poznat kao Java 1.6): Objavljeno 2006. godine, ovo izdanje je uvelo poboljšanja Java virtuelne mašine (JVM), uključujući podršku za dinamičke jezike i Java Compiler API.

Java 7: Objavljena 2011. godine, ova verzija je uvela naredbu try-with-resources, naredbe višestrukog hvatanja i podršku za nizove u naredbama switch.

Java 8: Objavljeno 2014. godine, ovo izdanje je dodalo podršku za lambda izraze, Stream API i Date/Time API.

Java 9: Objavljena 2017. godine, ova verzija je uvela Java Platform Module System (JPMS), koji omogućava modularno programiranje.

Java 10: Objavljeno 2018. godine, ovo izdanje je dodalo podršku za zaključivanje tipa lokalne varijable i interfejs za prikupljanje smeća (Garbage Collector).

Java 11: Objavljena 2018. godine, ova verzija je uvela podršku za HTTP/2 i predstavila nove standardne karakteristike Java Platform Module System (JPMS).

Java 12, 13, 14, 15, 16 i 17: Izdate 2019, 2020 i 2021, ova izdanja su donijela nove funkcije, poboljšanja i poboljšanja performansi Java platforme.

Kao što možemo zaključiti, svako sljedeće izdanje Jave donosi nam nove karakteristike, poboljšanja i unapređenja performansi platforme. Zbog toga se programeri ohrabruju da ažuriraju svoje Java instalacije kako bi iskoristili prednosti najnovijih funkcija.

## Instalacija Jave

[bla, bla, bla]

## Biblioteke

[bla, bla, bla]

# Sortiranje

[Uopšteno o sortiranju. Zašto nam treba]

## Algoritmi sortiranja

[O algoritmima sortiranja koji sve postoje]

## Quick sort

[bla, bla, bla]

## Shell sort

[bla, bla, bla]

# Paralelizam u algoritmima sortiranja

[Kako nam se čini. Ima li potrebe za tim. Jesu li se desila neka velika poboljšanja.]

# Paralelizam u Javi

[Da li je moguć? Kako se postiže? Šta dobijamo njime?]